CHAPTER

1

INTRODUCING LINUX

After reading this chapter and completing the exercises, you will be able to:

- Explain how an operating system works
- Explain how and why Linux was created
- Describe some benefits of using Linux
- Locate additional information about Linux commands and features

In this chapter you learn what an operating system is and how the Linux operating system compares to others that you may have already used. You learn about the unusual background of Linux and why many people feel so strongly about the way it continues to be developed. Finally, you learn about where you can find additional information regarding Linux.

UNDERSTANDING OPERATING SYSTEMS

If you are like most computer users, your experience is limited to working on one of the popular graphical computers such as an Apple Macintosh or a computer running Microsoft Windows 95, Windows 98, or Windows 2000. These platforms are popular and easy to use, but they aren't right for every computing need. Linux arose as an alternative choice for people whose computing needs require something other than the platforms most people are familiar with.

To appreciate the value of Linux, you must understand what an operating system is in relation to the computer itself and the programs (or applications) that you run on the computer. Then you can also understand the features and benefits that the Linux operating system offers.

Defining an Operating System

When computers were first created 40 or so years ago, everything the computer needed to do was **hard wired**, meaning that the instructions were arranged in the wires and other components that made up the computer. Because of this, a computer was able to complete only a single task—the one that it was hard wired to perform. For example, a computer

might be designed to add or subtract two numbers, but not have any other capability. As more powerful computer hardware became available, programmers began to demand more flexibility. The result was software. The term **software** refers to the programs that control the physical computer components, providing instructions for completing a task. Unlike a hard-wired system, software can be changed without disassembling the computer itself. Obviously, this was real progress.

Early software contained everything the computer needed to complete a task. Before long, however, programmers decided it was more efficient to create one kind of software that provided core functionality, and then to create applications that built upon those basic functions. An **application** is a program that provides a service to a person using the computer, rather than simply managing the computer's resources. For example, word processors and accounting software are applications. The core functionality was designed to send instructions to the hardware. Applications could then use the core functionality for more specific tasks. For example, one core function might be the ability to print characters on the monitor. An application could then use this functionality to print specific characters for a spreadsheet, a word processor, or other programs.

We call this set of core functionality the operating system. Although operating systems have advanced a great deal in the last 30 years, the basic purpose remains the same: an **operating system** is the program that contains a set of core functionality for other programs, providing both the interface between other programs and the hardware, and the interface between other programs and the user sitting at the computer. Figure 1-1 shows the relationship of the user, the application software, the operating system, and the computer hardware.

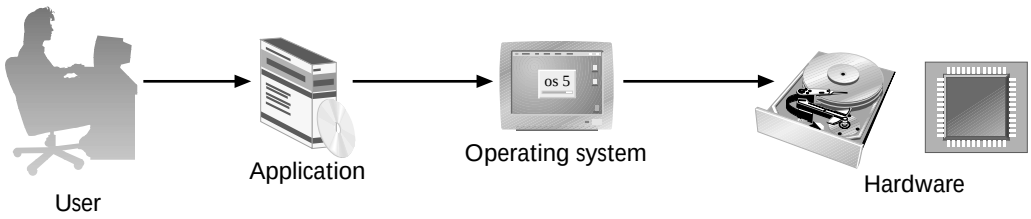

Figure 1-1 Relationship between the user, applications, operating system, and hardware

Operating System Functions

Today you can choose from many operating systems. Although they vary in appearance and functionality, they do have many similarities. For instance, all applications rely on the operating system for basic services; the operating system in turn controls the computer hardware, doling out resources for applications as necessary. At its most basic, an operating system will usually do the following:

- Initialize (or prepare) the computer hardware so that the operating system and other programs can function correctly
- Allocate system resources, such as memory and processing time, to the programs that are using the operating system

- Keep track of multiple programs running at the same time
- Provide an organized method for all programs to use system devices (such as the hard disk, printer, and keyboard)

The effectiveness with which an operating system handles these basic tasks determines its power. These tasks are controlled by the core of an operating system, called the kernel. When we think of an operating system, however, we normally think of the kernel plus some additional components. These major parts of an operating system include:

- **Kernel**: the core of the operating system, which schedules when programs can use computer resources and interfaces directly with core components of the computer hardware, such as memory and hard disks
- **Device drivers**: special software that provides access to additional hardware, beyond core device support provided by the kernel
- **Utility programs**: special software that helps manage the hardware and operating system features (as opposed to doing other types of work such as word processing)
- **Graphical interface**: the program that provides mouse-driven applications with menu bars, buttons, and so forth

Commonly Used Operating Systems

Some of the better known operating systems that are or have been used around the world are described here. Many other operating systems besides those mentioned here are still in use. This section only summarizes a few key steps in the last 40 years of operating system development as they relate to the emergence of Linux.

The **UNIX** operating system was created at AT&T Bell Labs (now part of Lucent Technologies) about 30 years ago by Ken Thompson and Dennis Ritchie. It was designed to control networked computers being shared by many users. UNIX development has continued since it was first introduced, and the operating system is currently sold by dozens of large companies, such as IBM, Hewlett-Packard, and Sun Microsystems. The Internet was developed on UNIX and is still based around the UNIX operating system.

The Disk Operating System (**DOS**) was created in about 1980. Despite very limited functionality, it gained widespread acceptance when IBM introduced the first personal computer, the IBM PC, in August 1981. DOS was designed to make efficient use of very limited hardware resources for a single user on one computer.

Neither UNIX nor DOS included a graphical interface of any kind. The operating system provided only character-based screens, which required the user to type a series of commands. After a time, several graphical interfaces were developed for each operating system. These graphical interfaces provided additional core functionality that other programs could draw upon. The leading graphical interface for UNIX was called the **X Window System**. This system is still in use today and is also used for graphical displays on the Linux operating system. The leading graphical interface for DOS became **Microsoft Windows**.

In 1984 Apple Computer introduced the **Macintosh** system, which integrated the operating system and the graphical interface so that users remained essentially unaware of the operating system. Rather than having to type complicated text commands, users saw only the graphical interface. The Macintosh was designed to make new users feel comfortable. Although the popularity of the Macintosh never equaled that of Microsoft Windows (perhaps because Macintosh computers were much more expensive than IBM-compatible PCs), the idea of hiding the operating system from novice users took root. In August 1995 Microsoft introduced the Windows 95 operating system, which integrated a copy of DOS and a copy of Windows (both updated many times since earlier versions). In this new version of Windows, the user had no need to know about the operating system that provided the functionality for all the Windows programs running on the system.

While all this was happening in the world of personal computers, businesses continued to use UNIX systems and other specialized operating systems. These operating systems were much more powerful than DOS, Windows, or Macintosh systems, but they were also very expensive and only ran on costly computer hardware.

Microsoft continued development of its operating systems by creating a business-oriented product called **Windows NT**. Unlike the earlier version of Windows, Windows NT did not include DOS as an underlying operating system. Instead, Windows NT was built on the VMS operating system, a business-oriented operating system that had been used for years on expensive minicomputers. (A minicomputer, which is a multiuser computer midway between a personal computer and a mainframe, cost between about $20,000 and $100,000 in the early 1990s.) Windows NT developers sought to include UNIX features to capture the market of business users who wanted more power than a Macintosh or Windows 98 system provided.

The Arrival of Linux

Into this fray, Linux arrived as a relative newcomer with a strange background. In 1991 a college student in Helsinki, Finland, named **Linus Torvalds**, began to create an operating system kernel as a school project. He wanted to use a UNIX-like operating system (UNIX has always been popular on many college and university campuses), but he couldn't afford his own UNIX system. Instead he began to clone, or duplicate, the functionality of a UNIX kernel for his IBM-compatible PC. This in itself would not be different from the efforts of many other students working to create something useful to save a few dollars. But the efforts of Linus Torvalds blossomed into something much bigger because of the work of many people with goals similar to his. (For a photograph of Torvalds, see Figure 1-2.)

The next section describes how Torvalds and others working around the world were able to finish a complete clone of the UNIX operating system by about 1993, and why that operating system, dubbed Linux in honor of Linus, has had such a surprising impact in recent years.

Figure 1-2 Linus Torvalds, originator of the Linux kernel

THE SPIRIT OF LINUX

Several factors were working in favor of Linus Torvalds as he sought to create an operating system for his PC:

- Finland had (and has) excellent Internet access, especially on its college campuses. This gave Torvalds access to a worldwide network of people who could help him develop Linux.
- As a computer science student, Torvalds (and those who helped him) could draw on 40 years of shared experience with the UNIX operating system. All of the design and experimentation that had gone into creating UNIX could be implemented in the kernel that Torvalds created from scratch.
- Torvalds was working as an individual, with other individuals. Decisions about how to do things could be based on technical considerations rather than on market needs, a factor that often drives the actions of commercial software companies, sometimes to the detriment of those using the software.
- The most popular operating system at the time (that is, from 1991 to 1996) was Microsoft Windows. Many Windows users were becoming frustrated with some of its features and with its lack of stability. They were also frustrated by delays in Microsoft's promised upgrades to Windows.
- Torvalds decided to base the software license for the Linux kernel on a model used by Richard Stallman and the Free Software Foundation.

The last item in this list may surprise you. The fact that the license to a piece of software would be so important in its history is unusual and bears more explanation.

The Linux Software License

A **software license** is a legal definition of who can use the software and how it can be used. The programmer who develops a piece of software decides how the software will be licensed. Licenses for commercial software usually state that you can use one copy of the software (the one you paid for), that you may not copy the software, that the company is not responsible for how you use it, and so forth. The license for Linux is quite different, as you will learn in the following sections.

The Free Software Foundation and the GNU Project

In 1983 **Richard Stallman** at the Massachusetts Institute of Technology founded an organization called the **Free Software Foundation (FSF)**. Stallman's motivating idea was that software should be freely available, without restrictions on copying. He proposed that companies could make money by charging for services and customization, but that the software itself should not be restricted in its distribution by a standard commercial license agreement. To back up his opinions, Stallman and those working with the FSF created hundreds of utilities that run on the UNIX operating system and distributed them freely around the world. This effort was called the **GNU project**. The GNU project intended to create a completely free UNIX-like operating system. When almost the entire project was finished, Linus Torvalds appeared with the final, crucial piece: the kernel. The software created by the GNU project is still used by millions of people and is included with every Linux distribution.

One of the best known products of the GNU project is the C language compiler called **gcc**. This is a software program for converting C language programming instructions into code that a computer can execute. The gcc compiler is the most widely used, highly regarded compiler in the world.

The explanation given here for the work of the Free Software Foundation is necessarily simplistic. Much more information about both the philosophy of free software and the relationship between GNU and Linux is available at *www.fsf.org*.

The GNU General Public License (GPL)

The license that Richard Stallman designed for the programs created by the GNU project is called the **GNU General Public License**, often abbreviated as the GPL. Torvalds eventually released the Linux kernel under the GPL. The GPL allowed Linux to develop rapidly, but it is an unusual license in several ways. It includes the following points:

- A programmer who decides to license a piece of software under the GPL gives away the source code to the software. The **source code** is the set of human-readable programming instructions used to create the program. Normally, only the machine-readable **binary code** used to execute a program is distributed. Including the source code makes it possible for anyone to modify the original program.

- Anyone can distribute the software, charging money for it if he or she chooses. But distributors must also include the source code, and they cannot restrict the redistribution by anyone else.
- Anyone who makes modifications to the original program must freely give away those modifications, including the source code.

Almost from the beginning, the term *copyleft* was associated with the GPL. The legal process called *copyright* has always protected creative work by authors, artists, musicians, and others. A copyright lets the creator of a work control how that work is used, so that others cannot rob a creative person of his or her livelihood by using creative works without permission. Conversely, the GPL does not let an author (that is, a programmer) *control* a creative work; instead it lets the author maintain *credit* for a work, while letting everyone benefit without charge for what the author has done. The term **copyleft** is used ironically to describe this radical departure from the customary copyright arrangement. Legally, the author of a GPL program still maintains the program's copyright. By choosing to release software under the GPL, the author does not choose to give up the copyright; instead, he or she is simply stating who can use the software and how. The author's name always remains associated with the software that he or she created.

The name **OpenSource** is often used to refer to software licensed under the GPL. Although the FSF makes distinctions between the GPL and the term OpenSource, in practice they are used synonymously in the Linux industry.

The following list summarizes some results of the GPL for Linux in particular and the GNU project software in general:

- The code is very high quality (that is, it has few bugs, or problems, and it runs efficiently) because every programmer who wishes to can make improvements to the software. The process of refinement is more rigorous, with more people involved, than in almost any commercial software company. Developers who create software using the GPL must be open to accepting suggestions for improvements from other skilled programmers.
- A program can be developed very quickly because it can build upon similar software already in use. For example, if hardware driver software is needed for a new video card, the drivers for existing video cards can be used as a starting point. New or different features for the new video card are really just modifications to existing software. Adding to the speed of development is the fact that hundreds of interested developers from around the world might be working on a particular project.
- Because the GPL license does not permit companies or individuals to keep improvements to themselves, each developer's work benefits everyone else. Hence, qualified people are willing to spend time developing software for public use,

because they know that other people are doing the same thing. Those who work on GPL software usually focus on something that is immediately useful in their own projects. They are willing to give their code away as a means of saying "thank you" for the work that others have given them.

Developers who have created Linux include not only highly skilled young people, from high schools as well as universities, but also computer science professors, researchers, system administrators for large companies, and hundreds of others.

The GPL was not the first license to allow free redistribution of software. Nor was it the last. Other similar licenses are used for the Berkeley version of UNIX (called FreeBSD), the Apache Web server, the X Window System servers included with Linux, and many other programs. The real value of the GPL for Linux is that it requires source code to be included with each program. It also requires that enhancements be given away in the same manner (that is, the GPL applies to enhancements to a GPL program).

People are often confused about what is *not* licensed under the GPL. A program that becomes part of the Linux kernel must abide by the GPL, because Linux abides by the GPL. But a program that *runs* on Linux is not part of Linux itself. Thus, Linux applications do not need to be released under the GPL. IBM, Corel, and other companies have begun to sell Linux applications. They could not do this if all Linux applications were regulated by the GPL.

Software called *system libraries* may be shared by both free programs and commercial programs operating in memory at the same time. The use of these libraries is covered by a version of the GPL called the Library GPL (or the **LGPL**) .

How Linux Is Developed

Linux kernel development follows the model of most GPL or OpenSource projects. To begin a project, a person identifies a need and then begins writing a program. At some point, the developer announces the work on the Internet. Developers who share an interest in that project respond, and soon they begin to work together on different parts of the project. This process works well and in fact is exactly how Linus Torvalds started to create the Linux kernel.

After a certain level of completion has been reached, the project's source code is released on the Internet. (Since the inception of Linux, the source code has been distributed via the Internet site *ftp://ftp.funet.fi*, based in Finland.) At this point, thousands of people download the source code and begin to try it out. Some of those people send back information about problems they have encountered (software bugs). The core team of developers tries to fix the bugs, occasionally working with other developers who have submitted bug fixes or specific enhancements to the software.

Linus Torvalds continues to work on the Linux kernel, along with a core group of developers who control what is included in Linux. Some commercial users of Linux have posed the unpleasant question of what will happen to Linux if something happens to Linus Torvalds. The answer is that another of the core developers on the Linux kernel team would simply take over. Each member of that team is a recognized expert in Linux. Torvalds leads them now, but he could conceivably retire from that position, leaving another qualified individual to continue the work.

Linux Distributions

The Linux kernel originally created by Linus Torvalds did not provide the functionality of a full-blown commercial operating system. To be really useful for the majority of organizations, Linux also requires:

- Networking utilities
- System administration utilities
- Documentation
- Installation tools
- Technical support information
- Hardware drivers
- A graphical environment (like the X Window System mentioned previously)
- Graphical tools
- Personal productivity applications such as word processors or spreadsheets

Given the way that Linux is developed (with developers all over the world creating and documenting its various parts), you might wonder how the many pieces of Linux could be combined into a complete operating system. Such a "productized" version of Linux, which includes many software components, installation tools, documentation, and so forth, is called a **Linux distribution**.

A Linux distribution has the Linux kernel at its core, along with hundreds or thousands of additional programs that run on Linux. Most of these are related to managing the Linux system—in other words, they are system utilities. These system utilities are drawn largely from the GNU project of the FSF. For this reason, many people refer to a Linux distribution as the GNU/Linux operating system. When taken as a whole, a Linux distribution makes it possible for nonprogrammers to install and use Linux.

Distributions in the Marketplace

Many new companies are attempting to meet the need for a Linux system that is ready to use. In accordance with the GPL, these companies include in their distributions the source code for the Linux kernel, as well as many other utilities. They can charge as much as they choose for their Linux distributions; because the software is freely available from other sources,

the distributor essentially functions as a kind of packaging service that saves users the trouble of downloading a large number of files from the Internet. Thus distribution prices have generally been quite low—between $2 and $100. Because the Linux kernel and utilities offered by the many Linux vendors are practically identical, vendors often add commercial components or other software to make their distribution more attractive to consumers.

Table 1-1 lists several popular Linux distributions, along with the Web location where you can learn more about these products. Each company tends to focus on a specific type of customer. For example, Caldera seeks to provide a business-oriented product. TurboLinux is aimed at those requiring specialized Linux servers. SuSE is a distribution with an international background that attempts to include many software components.

Table 1-1 Popular Linux Distributions

Name	Comments	Web site
Red Hat Linux	The most widely used distribution in the world, from Red Hat Software.	*www.redhat.com*
OpenLinux	Produced by Caldera Systems. This distribution is aimed at business users.	*www.calderasystems.com*
Mandrake	Built on Red Hat Linux with additions.	*www.linux-mandrake.com*
Stampede	A distribution optimized for speed.	*www.stampede.org*
Debian	A noncommercial Linux distribution targeted specifically to free software enthusiasts. Debian does not have a company behind it. It is created and maintained by developers of free software.	*www.debian.org*
Slackware	One of the first Linux distributions; still maintained by its original creator, Patrick Volkerding. Distributed by Walnut Creek CDROM.	*www.cdrom.com*
SuSE	The leading German distribution. Now available in the United States.	*www.SuSE.com*

Distributions and the Future of Linux Development

Some Linux developers claim that commercial Linux distributions exploit the work of unpaid Linux developers solely for commercial profit. Most Linux developers, however, are glad to see their work widely distributed and enjoy the prestige that comes from having millions of people using their software. These developers also recognize that large organizations cannot afford to take the risk involved in using software that has been freely downloaded from the Internet—even if that software is reputedly of very high quality. Furthermore, Linux companies like Slackware, SuSE, and many others make using Linux a viable commercial option for large organizations; they contribute money and personnel toward developing and refining free software.

You should also keep in mind that, while individuals are free to download, install, and upgrade software at will, organizations with hundreds of networked computers have to take

a more systematic approach. Among other things, organizations require an entity through which they can acquire products with guarantees of upgrades, technical support contracts, and similar standard business services. Commercial Linux vendors fill this role.

The competition of the marketplace also pushes Linux vendors to make better, more powerful, and easier-to-use products. This benefits all Linux users. Because the technology is controlled solely by the developers, Linux vendors have less control over the prices of their products than do large vendors of proprietary software. Thus Linux users can be sure they are paying a fair price for a particular distribution. As a rule, prices remain low because the various distributions differ only slightly. If the price were to rise too high, users would choose to download the software for free; or else another company would create a new distribution at a lower price. (In the language of business schools, the barrier to entry as a new Linux vendor is fairly low—anyone can create his or her own Linux distribution using the free software on the Internet.)

Version Numbering

The version numbers of Linux can be confusing. First of all, keep in mind that each release of the Linux kernel is assigned a version number. At the same time, different version numbers are assigned to each component of a Linux distribution; these version numbers are assigned by the developer in charge of that component. The Linux distributions themselves also have version numbers, which are chosen arbitrarily or for marketing reasons, based on how often the distribution is revised and updated.

The version number of the Linux kernel includes three parts:

- A major version number, which changes very rarely. Currently Linux is in major version 2.
- A minor version number, which changes infrequently, perhaps every 10–18 months. Even-numbered minor versions are stable operating systems that are used for creating commercial Linux distributions. Odd-numbered minor versions are development versions of Linux that should not be used except by experienced Linux users because they may crash at any time. Development versions of Linux are used as interim releases while a new stable version of Linux (with additional features) is being created.
- A patch level, which changes very frequently for development versions of Linux, perhaps once per day or once per week. For stable versions of Linux, this number changes only a few times as problems are located and fixed to make the stable Linux kernel even more solid.

A version number for the Linux kernel might look like this: 2.2.10. This is a stable release of the kernel (as indicated by the second 2). It is also patch level 10, indicating that the 2.2 kernel has had several minor updates for stability or improvements in hardware support. Another Linux kernel version might look like this: 2.3.67. In this case the second number is odd, indicating that it is a development release of the kernel, which should not be used in a business environment. The 67 indicates that 67 versions of this release have occurred, each with fixes or enhancements added by the kernel developers. After a certain number of enhancements have been added and

made stable, the kernel developers will decide to release a kernel version 2.4.0. The process then begins again as they work on adding new features with the 2.5 series of kernels.

Don't run the development kernels on your servers unless you want to experiment with new features that might crash your system. Commercial Linux distributions always use stable Linux kernel releases.

Linux distributions created by companies such as Red Hat Software and Caldera Systems are composed of hundreds of programs. The version of the Linux kernel included with a product is important, but a separate version number for each component could also be mentioned, for example, the version number of the Apache Web server or the version of the gcc compiler. To avoid the need to specify which version of each component is included in a distribution, vendors of commercial Linux distributions assign a version number to the distribution as a whole. For example, Red Hat Linux 6.1 contains a 2.2.12 Linux kernel. The 6.1 designation for the distribution corresponds to the vendor's own numbering system, and has no real relation to the kernel number. The Linux distribution released by Caldera Systems about the same time was numbered 2.3, though it had almost the same set of software.

Don't worry much about the version number when comparing distributions. Instead, look at the versions for the individual components that matter most to you, including the Linux kernel, the graphical system, and specific services you need to use. You can get information about all these specific version numbers from the vendor that created a Linux distribution.

The Motivation of Free Software Developers

As Linux is discussed in business and computer publications, those unfamiliar with the software development model used by Linux and other free software ask the same question: "Why would so many people devote so much effort to something without expecting any reward? It just doesn't make sense."

Of course, the business world that asks these questions is driven by money, and when considering money alone, free software development does not make much sense. Linux developers, however, are driven by other motivations. After reading the first half of this chapter, you can probably compile a list of these motivations yourself. The following summarizes the forces that motivate many developers of Linux and other free software:

- Creating a piece of software often fills a developer's specific technical need. By creating the product as free software, the developer can effectively thank others for the free software that has already proven useful in his or her work.
- Within the free software community of developers, those who create the highest quality, most original work are regarded very highly by their peers. The respect of like-minded professionals whom you respect in turn is a powerful motivating factor. In addition, if you create a unique, powerful product, many people will use it, see your work, and thank you for it.

- The Linux community and other similar communities revolving around products like the Apache Web server are very popular in the news, with an increasing visibility and acceptance by major organizations. Participating in free software development gives a sense of contribution and community to developers, many of whom are not highly social, but nevertheless like the opportunity to be a part of something worthwhile. As the market strength of Linux continues to grow, participation in a well-known free software project is also a tremendous boost to a resume.

After reviewing this list, you might be thinking, "That's fine for some people, but I still need to earn a living." In that case, you'll be happy to learn that Linux can set you on the path toward a fulfilling and profitable career. Here is a list of occupations open to those with a strong understanding of Linux and its related technologies:

- Software engineer: many companies are using Linux as a development platform or Internet server and need qualified programmers to create software to run on Linux. The information in this book provides a good first step towards becoming a Linux developer.
- System administrator: each of the thousands of companies that uses Linux needs at least one qualified system administrator to keep the Linux server running smoothly, day after day. This book is designed to prepare you to be a Linux system administrator.
- Trainer: as more companies begin using Linux, they require training for both technical and nontechnical specialists who will be using Linux to complete their work. Trainers work in all types of companies, teaching people how to use Linux.
- Writer: if you are a good writer, you can share your knowledge about Linux with others. You might consider a career as a technical writer for a company, as a writer for a business-oriented computer publication, or some combination of these two. Keep in mind that new periodicals, books, and columns about Linux are appearing every month.

Even if you are not seeking a job in a field directly related to Linux, you should know that many companies now recognize Linux expertise as a sign of generally strong computer knowledge. Such companies list knowledge of Linux as a qualifying skill for many types of jobs, including training, marketing, sales, and technical management.

The following lists some Web sites that you can visit to research jobs related to Linux. To learn about Linux-related jobs, just search on the keyword *Linux* at any of these Web sites, or refine your search criteria further based on your specific interests. Each of these sites will give you a sense of the promising future that awaits students of Linux.

- *www.dice.com*
- *careers.wsj.com*
- *www.headhunter.net*
- *www.careermosaic.com*

THE STRENGTHS OF LINUX

In the previous sections you learned about the background of Linux and the license under which it is distributed. These are certainly contributing factors to the ongoing popularity of Linux. But the reason millions of people and many large organizations use Linux is not because it is free. They use Linux because it is a high-quality operating system. The next sections describe some features of Linux that make it so popular.

Stability

Linux has proven its stability in many organizations. Many businesses have run a Linux server continuously for more than a year at a time without any problems and without the need to reboot (restart) the system. This stability is in part due to the fact that Linux can end a program without affecting other programs or the operating system as a whole. Another reason for its stability is that the core functionality of Linux, such as how system memory is used, how the hard disk is accessed, and how programs share system resources, has been thoroughly tested by the thousands of people involved in each version of Linux.

Security

The same development process that yielded a highly stable operating system also yielded a very secure operating system. You might be tempted to conclude that an operating system with freely available source code could not possibly be secure. On the contrary, the fact that the source code is available to all, and released in a controlled manner by well-known, respected professionals, means that all interested developers can help identify and fix security problems.

Proponents of other operating systems may point to the fact that many security holes have been identified on Linux systems. In truth, these security problems almost always relate to programs running on Linux, such as the `sendmail` e-mail server program. And in any case, when users do discover security problems, Linux developers will create a software update to fix the problem within about 24 hours. These updates are posted on the Internet so that all Linux users can download them. In contrast, developers of other operating systems seek to create secure systems by hiding the security problems that have been identified. They do not generally provide open information and rapid software updates to fix potential problems.

To learn more about security on computer systems, visit the Computer Emergency Response Team site at *www.cert.org*.

Speed

Linux was designed to use limited hardware resources efficiently. As a result, Linux makes better use of hardware resources than almost any other operating system. As recently as 1996, a complete Linux operating system could be run on a system with only 4 MB of system

memory. The efficiency of Linux when operating with such limited resources translates into speed when more extensive resources are available. Given a certain piece of computer hardware (such as a Pentium computer running at 400 MHz with 64 MB of RAM), you will see better performance from Linux than from any other general-purpose operating system.

A Multitasking, Multiuser, Multiprocessing System

Linux is a true multitasking operating system, which means that it can run many programs at the same time. (A typical Linux system will have 20 to 50 programs running at the same time.) Although many operating systems can run multiple programs simultaneously, Linux does this very efficiently. What's more, a program can crash in Linux without affecting the other programs running on the system. This helps to create a stable system.

Linux manages multiple programs through a technology called **preemptive multitasking**, in which the Linux kernel controls which program runs at any given moment. Once a program has had a small time to work, the kernel intervenes and gives control to another program for a time. By contrast, some operating systems use **cooperative multitasking**, in which the kernel is forced to wait for a program to yield control. Cooperative multitasking can be problematic because it is possible for a poorly written program to crash before the kernel can regain control of the system, thus disabling all other programs running on the system.

Linux was also designed from the beginning as a **multiuser system**, which means that multiple users can log in to the same Linux system over a network connection and run programs, use the Internet, or complete other work. The programs run by one user do not affect the work of other users. A superuser, or administrative account, can configure and control all user accounts.

As a security feature, users must log in to a Linux system (using a valid username and password) before attempting to do any work.

Linux also supports multiple CPUs on the same computer, thanks to a technology called **symmetrical multiprocessing**. Systems with multiple processors perform faster than single-CPU systems because the processors can combine forces to work on one task at the same time. Linux (and some other operating systems) divide the components of a task between multiple processors via a technique called **multithreading**, in which a program is divided into parts, known as threads; the various threads of one program are then run simultaneously on multiple processors. Multithreading is found only in more advanced operating systems such as Linux, Windows NT, and UNIX. On commercial operating systems, using multiple processors adds significantly to the cost of the operating system. In contrast, all versions of

Linux provide multiprocessing capability. Figure 1-3 illustrates the concept of a multiuser, multitasking, multiprocessing operating system.

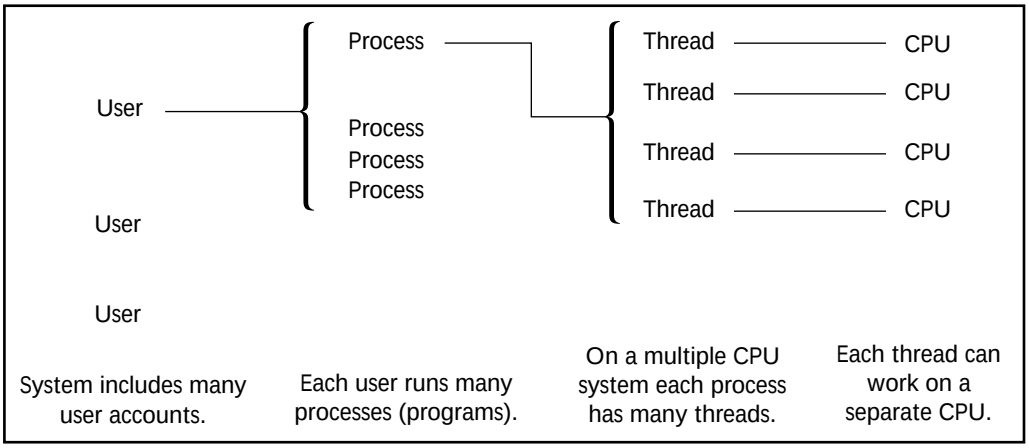

Figure 1-3 A multiuser, multitasking, multiprocessing operating system

Flexibility

Linux distributions are extremely flexible because they always include the source code to the operating system, allowing technically oriented system administrators and software developers to modify a system any way they want. More in-depth modifications may require outside help, but Linux developers are easily consulted via the Internet, and they are generally more than willing to help.

By comparison, operating systems that do not include source code offer little in the way of flexibility. With these systems, administrators are limited to asking the operating system manufacturer for an update that meets a specific need. And even then, the answer to such a request is usually, "Sorry, we can't do that."

Although you are free to modify a Linux system as much as you want, you certainly don't have to make changes to keep your system running smoothly. Linux is flexible enough to allow you to use old, stable technology that fulfills the job at hand, or to experiment with the latest advances and features. One Canadian retail organization that chose the first approach has been running a copy of the Linux kernel version 1.2.35 for years without a hitch and has no plans to upgrade to a new system.

Contributing to the flexibility of Linux is the fact that with only a few modifications, you can often run newer Linux applications on older versions of the kernel. With other operating systems, you would probably have to completely install a new version of the operating system before you could use the latest programs.

Some organizations might opt for a more restrictive, commercial operating system because they are intimidated by the flexibility of Linux and all the options it provides. But in fact, commercial Linux vendors such as SuSE and Red Hat Software provide the technical support commonly associated with commercial operating systems; thus, management teams can be confident that a Linux expert is only a phone call away. (Technically oriented individuals might prefer the Internet resources described later in this chapter.)

Applications

Years ago, Linux was used almost exclusively for developing UNIX software or for specialized Internet servers. Back then, critics of Linux might argue, "What good is an operating system that doesn't run the applications I need?"

Today that concern is unfounded. Although Linux still finds its greatest concentration of supporters among those running Internet servers and developing software, the number of applications available for Linux continues to grow rapidly and already includes the most prominent names in the software industry. Table 1-2 lists some of the programs currently available for Linux, as well as Web addresses where you can find more information.

Table 1-2 Common Linux Applications

Application	Description	Web site
WordPerfect for Linux	Complete, powerful word processor; from Corel	*linux.corel.com*
ApplixWare	Complete office suite	*www.applix.com*
StarOffice	Office suite and integrated Internet tools; attempts to imitate Microsoft Office	*www.sun.com/staroffice*
DB2	Powerful database package; from IBM	*www.ibm.com*
Oracle	The most widely used client/server database	*www.oracle.com*
Sybase	A popular client/server database package	*www.sybase.com*
Informix	A popular client/server database package	*www.informix.com*
UniCenter TNG	A graphical management console for very large networks; from Computer Associates	*www.cai.com*

Learning More About Linux

This book contains an organized introduction to all the important topics you need to understand in order to use Linux effectively. You can explore additional Linux topics through a number of different venues, as described in the remainder of this chapter.

Reading Linux Documentation

Because the developers of Linux were working entirely via the Internet, they were forced to share descriptions of their software via electronic or online documentation. This documentation was typically incorporated into the various distributions. These days most Linux distributions

include thousands of pages of online documentation. As you will see, this documentation can be quickly accessed using a few simple commands.

When you read Linux documentation, remember that much of it was written by developers, for developers. That is, these documents generally assume a high level of technical understanding. They contain a lot of useful details, but may be hard for a beginner to understand. As you begin investigating the world of Linux documentation, don't expect to understand everything all at once. Once you become more experienced with Linux, you'll find the documentation more useful.

The Linux Documentation Project

The **Linux Documentation Project (LDP)** was begun by Matt Welsh in the early 1990s, when Linux was just becoming well-known. The LDP was one of the first efforts to document how Linux works and continues to provide Linux documentation today in a variety of formats. The LDP currently contains about 6000 pages of documentation, all of which is available free of charge under a version of the GPL modified for use with documentation. The LDP consists of several types of documents. Some of these, such as the *Network Administrator's Guide* and the *kernel Hacker's Guide*, are complete online reference manuals.

Initially you will probably find the LDP's HOWTO documents most useful. The **HOWTOs** cover specific topics, such as sharing a system between Windows NT and Linux, or maintaining network security. (Figure 1-4 shows a sample HOWTO document.) Each document is written by one person, or a small group, with expertise in that topic. Documents called mini-HOWTOs focus on narrower subjects than do regular HOWTOs. HOWTO documents are usually written by software developers, but they are intended for all users, not just other developers.

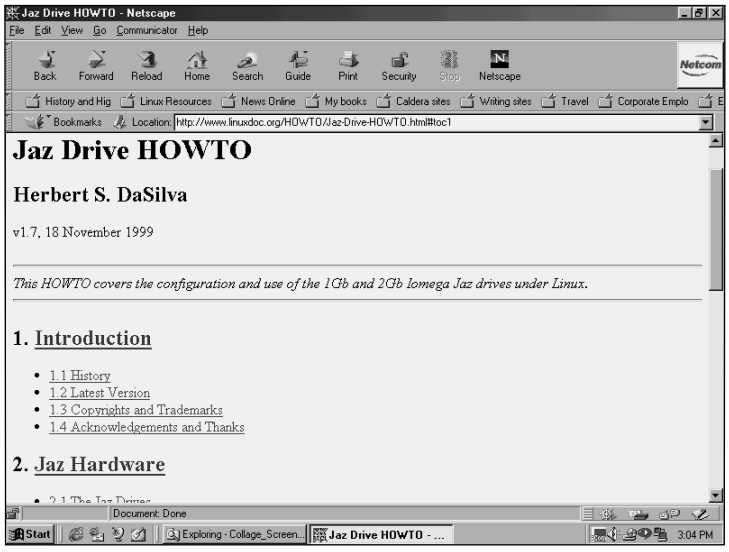

Figure 1-4 A sample HOWTO document

The LDP and all of the HOWTO documents are included with all Linux distributions. You can also read them on many Web sites. A good place to begin researching the documentation available as part of the LDP is *www.linuxhq.com/info.html*. The LDP is also available at *www.linuxdoc.org*. Table 1-3 lists some sample HOWTO documents.

Table 1-3 Sample Linux HOWTO Documents

Title	Description
CD Writing HOWTO	Using Linux with a writeable CD-ROM drive
Chinese HOWTO	Making Linux work effectively with Chinese characters
DOSEMU HOWTO	Using the DOS emulator in Linux to run DOS programs
Kernel HOWTO	Upgrading and modifying the Linux kernel
Laptop HOWTO	Installing and using Linux on a laptop computer
MP3 HOWTO	Downloading, playing, and creating MPEG3 audio files on Linux
Security HOWTO	Creating a more secure Linux system
Large Disk (mini-HOWTO)	Making optimal use of very large hard disks within Linux
ZIP Install (mini-HOWTO)	Installing an Iomega ZIP drive to be used in Linux

Linux Command Information

As you will learn in future chapters, Linux includes many different commands, each with numerous options. Few people memorize all of the commands and options, because they employ only a small percentage of them regularly. When you do need to use a new command, you can learn about it through the Linux command information available online. This information is provided in two formats. You can access online manual pages (called **man pages**) for most Linux commands by using the `man` command. For some Linux commands, the definitive source of information is an info page, which you can view using the `info` command. These commands will be described in detail in later chapters.

Documentation Included with Software Packages

Most of the software packages included with a Linux distribution provide at least some documentation. This documentation is installed on your system along with the software and you can usually view it in either a text editor or a Web browser. You will learn more about reviewing product or package documentation in future chapters.

Linux on the Internet

Linux was created on the Internet, and the Internet is still a great place to find out more about Linux. Every day on the Internet, developers release new software, companies make announcements about Linux products, and software documenters provide new or revised

information. Table 1-4 lists some Web sites that you will find useful as you explore the world of Linux.

Table 1-4 Linux-Related Web Sites

Web site	Description
www.linuxjournal.com	A high-quality companion Web site to the monthly printed magazine; contains additional Linux links and information (mostly technically oriented)
www.linuxworld.com	A business-oriented online magazine with interviews, links, technical reports, and other up-to-date information
www.slashdot.org	An eclectic collection of news items related to free software and other topics (such as Star Wars, new music technologies, cryptography legislation, etc.) of interest to free software developers
www.linuxhq.com	A collection of information about work on the Linux kernel, with useful links to many other sites and Linux resources
www.linuxapps.com	Links to information about applications that run on Linux
www.lwn.net	Linux Weekly News, a collection of news items related to Linux and other free software

The Web sites for each of the Linux distributions (listed in Table 1-1) are also great resources for learning about Linux. You will learn about additional Web sites in later chapters.

CHAPTER SUMMARY

- An operating system provides an interface between the computer hardware and the applications run by the user. In its most basic form, an operating system manages the use of memory, CPU time, and other system resources. A complete operating system includes many other features that provide additional hardware support, such as a graphical environment and driver software.
- The Linux kernel was created by many talented individuals from around the world working under the leadership of Linus Torvalds, who continues to maintain the Linux kernel. The Free Software Foundation, led by Richard Stallman, created hundreds of software programs as part of its GNU project. These are included with the Linux kernel in each copy of a complete Linux operating system. The General Public License is responsible in large part for the phenomenal growth of Linux in the last few years.
- Various companies have created commercial products, called distributions, that are built around the Linux kernel and GNU software. Businesses are beginning to value knowledge of Linux highly as more people recognize the features that Linux offers, such as stability, speed, flexibility, and low cost.
- Information about Linux is available online as part of the Linux Documentation Project, which includes many HOWTO documents on specific topics. Online documentation

for Linux commands is included with every copy of Linux. Many publications and Web sites maintain daily news updates about what is happening in the world of Linux and free software.

KEY TERMS

application — A program (such as a word processor or spreadsheet) that provides a service to a person using the computer, rather than simply managing the computer's resources.

binary code — Machine-readable instructions used to execute a program.

cooperative multitasking — A technique in which an operating system kernel must wait for a program to yield control to other programs.

copyleft — An ironic term that refers to the GNU General Public License (the GPL), signifying a radical departure from standard copyright.

device drivers — Software that provides access to additional hardware, beyond core device support provided by the kernel.

DOS — An operating system developed for personal computers in about 1980. It gained widespread acceptance when IBM introduced the first IBM PC.

Free Software Foundation (FSF) — An organization founded by Richard Stallman to promote his ideals of freely available software and to create and distribute that software.

gcc — A C language compiler. Probably the best known product of the GNU project.

GNU General Public License (GPL) — The free software license that Richard Stallman of the Free Software Foundation developed for the programs created by the GNU project.

GNU project — An effort by the Free Software Foundation to create a free UNIX-like operating system. Much of a Linux distribution comes from the GNU project.

graphical interface — Software that provides mouse-driven applications with menu bars, buttons, and so forth.

hard wired — Computer functionality that is arranged in the wires and other components that make up a computer. Hard-wired functionality cannot be easily altered.

HOWTOs — Documents within the Linux Documentation Project that cover specific topics.

kernel — The core of the operating system, which interacts directly with the computer hardware.

LGPL — A special version of the GNU General Public License intended to govern both free and commercial software use of software libraries.

Linux distribution — A Linux operating system product that includes the Linux kernel plus many software components, installation tools, documentation, and so forth.

Linux Documentation Project (LDP) — One of the first efforts to document how Linux is used. Started by Matt Welsh.

Macintosh — A computer developed by Apple Computer that integrated the operating system and the graphical interface.

man pages — Online manual pages for Linux commands. The man pages are accessed using the `man` command.

Microsoft Windows — The leading graphical interface for DOS.

multithreading — A technique used within multiprocessing operating systems to divide a larger task between multiple processors.

multiuser system — An operating system on which numerous users can log in to the same computer (usually over a network connection).

OpenSource — A trademarked name often used to refer to software licensed under the GPL.

operating system — Software that provides a set of core functionality for other programs to use in working with the computer hardware and interfacing with the user running the computer.

preemptive multitasking — A technique used by the Linux kernel to control which program is running from moment to moment.

software — Instructions that control the physical computer components, but which can be changed because they reside on a changeable media such as a hard disk.

software license — A legal definition of who can use a piece of software and how it can be used.

source code — A set of human-readable programming instructions used to create a piece of software.

Stallman, Richard — Founder of the Free Software Foundation and the GNU project.

symmetrical multiprocessing — A technique that allows an operating system to support multiple CPUs on the same computer.

Torvalds, Linus — Originator of the Linux kernel; formerly a student in Helsinki, Finland.

UNIX — An operating system created at AT&T Bell Labs (now part of Lucent Technologies) about 30 years ago by Ken Thompson and Dennis Ritchie. UNIX is still widely used, and it provided the technical basis for Linux.

utility programs — Software that provides assistance in managing the hardware and operating system features (as opposed to doing other types of work such as word processing).

Windows NT — A business-oriented operating system product developed by Microsoft. Windows NT is not based on DOS as an underlying operating system.

X Window System — A graphical software environment used by almost all UNIX and Linux operating systems.

REVIEW QUESTIONS

1. Explain the difference between a computer that is solely hard wired and one that uses an operating system.
2. An operating system does *not* do which of the following:
   a. Allocate system resources such as memory and CPU time
   b. Initialize computer hardware so it can be used by software running on the computer

c. Keep track of multiple programs running at the same time

d. Provide word processing features for users

3. The kernel of an operating system includes a graphical interface. True or False?
4. Which of the following operating systems included a graphical interface when first released?
   a. UNIX
   b. Macintosh
   c. Linux
   d. DOS
5. Linus Torvalds began to create Linux because:
   a. he was hired as an operating system consultant by a major corporation.
   b. he wanted a powerful operating system but could not afford one.
   c. his professor required that each student create a basic operating system.
   d. he felt it would be a good career move.
6. The Free Software Foundation is dedicated to the idea that:
   a. no company should be able to charge for any software.
   b. the real value of software was in customization, not in selling mass-produced copies.
   c. Richard Stallman's C compiler was the best in the world.
   d. Linux was an important development in operating systems.
7. The GNU project is important to Linux because:
   a. it provides the majority of the system utilities used by Linux.
   b. GNU software is the only software compatible with Linux.
   c. the media attention generated by the GNU project has made Linux popular.
   d. Richard Stallman is a strong supporter of the Linux movement.
8. The GPL includes all of the following *except*:
   a. GPL software must include source code.
   b. modifications to GPL software must be given away.
   c. software that runs on a GPL operating system must be given away.
   d. a company can charge money for GPL software.
9. Explain why the term "copyleft" came to be used as an ironic comparison to "copyright."
10. In general usage, the name OpenSource software is synonymous with software released under the GPL. True or False?
11. The fact that the use of Linux is governed by GPL means that:
    a. Linux is of high quality because everyone can review and improve the code.
    b. Linux is low cost because Linus Torvalds does not allow any commercial interference.

c. Linux is low cost because most nations enforce the GPL commercial restrictions.

d. Linux development advances quickly because people around the world participate in its growth.

12. Why might Linux provide better security than operating systems that do not provide source code to users?

13. Name five things that a Linux distribution vendor might add to the Linux kernel when creating a product to sell.

14. Market dynamics (competition) keep the price of Linux low because it can be freely downloaded. True or False?

15. In the Linux kernel version 2.4.10, the second digit, 4, indicates:

a. a major kernel release number

b. a minor kernel release number for a stable kernel

c. a minor kernel release number for a development kernel

d. a patch release number

16. Version numbers for Linux distributions generally match that of the Linux kernel itself. True or False?

17. Which of the following is not a likely motivating factor for those who develop free software?

a. Greed

b. Altruism or thanks

c. Peer acceptance

d. Desire for interesting work

18. Name five Linux distributions and comment on any specific purposes or background for each one.

19. Which of the following statements is *not* true?

a. More businesses are using Linux and related products to run their businesses.

b. Writing free software teaches marketable programming skills.

c. Thousands of servers running Linux require competent system administrators.

d. Linux is not based on any other operating system.

20. Linux systems have been known to run for months or years without crashing. True or False?

21. Compare preemptive multitasking to cooperative multitasking and multithreading.

22. Name five major applications that are available for Linux.

23. The use of a document produced as part of the Linux Documentation Project is governed by standard copyright notice. True or False?

24. The _____ and _____ commands provide information about Linux commands.
    a. `HOWTO` and `mini-HOWTO`
    b. `man` and `info`
    c. `http` and `ftp`
    d. `GPL` and `LGPL`
25. HOWTO documents discuss a variety of specific subjects but are only intended for very advanced users. True or False?

HANDS-ON PROJECTS

Project 1-1

In this project, you review several sources of documentation for Linux. You may want to save the results of this project for future use. To complete this project, you should have a computer with access to the Internet and a functioning Web browser.

1. Open your Web browser and connect to the Internet.
2. Go to the homepage of the Linux Documentation Project, at *www.linuxdoc.org*.
3. Click on the **LDP Mirrors** link to see a list of sites that also contain a copy of the LDP. Find the site that is physically closest to you. (If you have a reason to believe that you would have faster access to a different site, choose it instead.)
4. Note how frequently the mirror site is updated, and then click on your chosen mirror site link.
5. Save the information on your mirror site for future access. For example, choose the **Bookmark** option in your Web browser to record the Web address of the mirror site.
6. Review the home page of the LDP mirror site until you find the HOWTO link.
7. Explore the HOWTO documents. What different formats are available?

Project 1-2

In this project, you look at some of the different Linux HOWTOs. To complete this project, you should have a computer with access to the Internet and a functional Web browser.

1. Open your Web browser and connect to the Internet.
2. Go to the mirror site for the Linux Documentation Project that you selected in Project 1-1.
3. Browse the titles of the HOWTO documents.
4. Open up a HOWTO document that interests you. Review the table of contents.
5. Read one section of the HOWTO and summarize its main points.

Project 1-3

In this project, you review the Web sites for several Linux distributions. To complete this project, you should have a computer with access to the Internet and a functional Web browser.

1. Open your Web browser and connect to the Internet.
2. Visit the following Web sites: *www.debian.org*, *www.calderasystems.com*, *www.suse.com*, *www.redhat.com*.
3. For each Web site, answer the following questions:
   a. Can you determine the focus of the Linux distribution? To whom is the company trying to sell products?
   b. What key features of its distribution does each group highlight?
   c. What supporting documents (such as magazine articles) can you find on the Web site?
   d. Which Web site and distribution appeal most to you and why?
4. Locate information about signing up for an e-mail list on each Web site. What is the purpose of these mailing lists?

CASE PROJECTS

1. You work at ColTech Limited as a system administrator. Your boss is planning to purchase a new Web server, for which she is considering several platforms, including an IBM UNIX system, Windows NT, and Linux. You prefer Linux, but your manager hesitates because of concerns about the way Linux is developed. She asks you to explain why Linux is more stable and secure than other systems, and how the company can run the new Web server on free software. Write a brief report summarizing your thoughts on the matter. What useful information might you find on the vendor Web sites in Project 1-3? What concerns might remain unresolved?
2. Your manager has started to use Linux as the new Web server, and things are going well. You need to create a new piece of software to add a feature to your Web server. You are considering making it available to others under the GPL. What justification would you give your employer (who is paying you to write the software) that it should be given away to others under the GPL? What concerns might your employer have about doing that? Under what circumstances do you feel that a piece of software should not use the GPL?
3. Do you anticipate problems in the future as Linux becomes more commercial and popular? How might this affect the attitude of free software developers? How would you feel about the increasing popularity of Linux if you were participating in Linux development? What could commercial Linux vendors do to help alleviate potential problems?